AFRL-VA-WP-TR-2004-3013

TURBULENT SENSITIVITY ANALYSIS FOR ENHANCING FUTURE AIRCRAFT

Andrew G. Godfrey

AeroSoft, Inc.
1872 Pratt Dr., Suite 1275
Blacksburg, VA 24060

OCTOBER 2003

Final Report for 17 October 2001 – 21 October 2003

THIS IS A SMALL BUSINESS INNOVATION RESEARCH (SBIR) PHASE II REPORT

STINFO FINAL REPORT

AIR VEHICLES DIRECTORATE
AIR FORCE MATERIEL COMMAND
AIR FORCE RESEARCH LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542

NOTICE

USING GOVERNMENT DRAWINGS, SPECIFICATIONS, OR OTHER DATA INCLUDED IN THIS DOCUMENT FOR ANY PURPOSE OTHER THAN GOVERNMENT PROCUREMENT DOES NOT IN ANY WAY OBLIGATE THE U.S. GOVERNMENT. THE FACT THAT THE GOVERNMENT FORMULATED OR SUPPLIED THE DRAWINGS, SPECIFICATIONS, OR OTHER DATA DOES NOT LICENSE THE HOLDER OR ANY OTHER PERSON OR CORPORATION; OR CONVEY ANY RIGHTS OR PERMISSION TO MANUFACTURE, USE, OR SELL ANY PATENTED INVENTION THAT MAY RELATE TO THEM.

THIS REPORT HAS BEEN REVIEWED BY THE OFFICE OF PUBLIC AFFAIRS (ASC/PA) AND IS RELEASABLE TO THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS). AT NTIS, IT WILL BE AVAILABLE TO THE GENERAL PUBLIC, INCLUDING FOREIGN NATIONALS.

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION: "TURBULENT SENSITIVITY ANALYSIS FOR ENHANCING FUTURE AIRCRAFT"

/s/

Victor S. Burnley
Project Engineer
Applied Computational Research
Computational Sciences Branch

/s/

Matthew J. Grismer
Leader
Applied Computational Research
Computational Sciences Branch

/s/

Douglas C. Blake
Branch Chief
Computational Sciences Branch
Air Vehicles Directorate

COPIES OF THIS REPORT SHOULD NOT BE RETURNED UNLESS RETURN IS REQUIRED BY SECURITY CONSIDERATIONS, CONTRACTUAL OBLIGATIONS, OR NOTICE ON A SPECIFIC DOCUMENT.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	**2. REPORT TYPE**	**3. DATES COVERED *(From - To)***
October 2003	Final	10/17/2001 – 10/21/2003

4. TITLE AND SUBTITLE
TURBULENT SENSITIVITY ANALYSIS FOR ENHANCING FUTURE AIRCRAFT

5a. CONTRACT NUMBER
F33615-99-C-3004

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
65502F

6. AUTHOR(S)
Andrew G. Godfrey

5d. PROJECT NUMBER
A059

5e. TASK NUMBER

5f. WORK UNIT NUMBER
0A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AeroSoft, Inc.
1872 Pratt Dr., Suite 1275
Blacksburg, VA 24060

8. PERFORMING ORGANIZATION REPORT NUMBER
6

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Vehicles Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7542

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/VAAC

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-VA-WP-TR-2004-3013

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
This is a Small Business Innovation Research (SBIR) Phase II report. Report contains color.

14. ABSTRACT

The sensitivity-equation method has been extended to accommodate three-dimensional, unstructured mesh topologies. The method is developed and demonstrated for inviscid, laminar and the two-equation turbulence models. Performance is enhanced by implementing a MPI parallel capability as well as a graphical user interface. For all cases, solutions to the linear sensitivity equations require less than 10\% of the computational time required to solve the non-linear flow problem. A central-difference utility is developed to assess the accuracy of the SEM results. The product is developed with the goal of being compatible with any flow solver --whether structured or unstructured --as part of a larger design environment.

15. SUBJECT TERMS
Sensitivity Equations, Unstructured meshes, CFD, Turbulent flow

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT:**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** (Monitor)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Victor S. Burnley
Unclassified	Unclassified	Unclassified	SAR	32	**19b. TELEPHONE NUMBER** *(Include Area Code)* (937) 904-4046

Contents

List of Figures

Chapter 1

Introduction

1.1 Identification of the Problem

Through ten aerodynamics Technology Effort Objectives (TEO), the Air Force Research Laboratory (AFRL) has a vital interest in supporting significant advancement in aircraft performance, mission effectiveness, and accurate engineering design. The Center of Excellence in Computational Sciences, as part of the Aeronautical Sciences Division (VAA) at AFRL, has set a goal to "develop and apply state-of-the-art computational simulation methods for design and analysis of air and space vehicles". By providing innovative sensitivity software, the focus of this SBIR effort has been to assist the AFRL Computational Sciences Branch in its goal to "provide highly efficient, accurate and affordable techniques to support the needs of [its] customers" [5].

The heart of this SBIR project is the Sensitivity Equation Method (SEM) – a much more efficient way to compute flow sensitivities than the more expensive finite-differencing approach. One can view the SEM as a stand-alone analysis quite separate from the actual flow solution. The flow variables enter the sensitivity problem as spatially varying coefficients in a linear partial differential equation (PDE). This linear PDE describes the sensitivity of the flow to a user-specified parameter (for example, Mach number, wing twist distribution, or angle of attack). Being linear, the SEM approach provides sensitivities at a fraction of the CPU cost of another flow solution.

One can employ any flow solution in the SEM no matter how obtained so long as we can reasonably construct the necessary coefficient data to solve the sensitivity PDE. Thus, the sensitivity software can be used with any CFD flow solver: commercial, public domain, or research-level. This is important because the sensitivity package is designed to be a credible part of a larger system.

The purpose of the Phase I and Phase II aspects of the project have been to build a solid numerical foundation on which to develop high-fidelity modeling (*e.g.,* turbulence modeling, aeroelasticity, and multi-body formulation). The "Plus-Up" effort has extended that work to unstructured meshes. A brief review of the accomplishments in Phase I and Phase II and their relationship to the Plus-Up effort is given below.

1.2 Relationship to Phase I

Beginning in 1998, the Phase I showed that the individual elements of a sensitivity package could be combined to determine discrete solutions to the continuous sensitivity equations. A brief summary of the Phase I accomplishments and benefits are given below.

1. For sensitivity problems needing high spatial accuracy, a *consistent* high-order Jacobian matrix was implemented to maximize the efficiency of computing the sensitivity solutions. Previous work used a first-order Jacobian regardless of the spatial accuracy in the flux scheme. Gains in efficiency for a single sensitivity solution are especially important when a large number of design variables exist in the problem.

2. Sensitivity solutions for multiple design variables became a possibility. This capability enhances the software's ease-of-use by providing flow sensitivities for the entire problem during a single run.

3. A formulation of the sensitivity problem for body-rate design parameters (for example, roll rate, pitch rate and yaw rate) was completed. Earlier work on force and moment sensitivities was enhanced by this body-rate capability. This step addresses interests in flight-dynamic stability and performance calculations.

1.3 Relationship to Phase II

After the Phase I, the envisioned sensitivity product – called $\mathcal{SENSE}$ – had matured to the level of inviscid or laminar flow, but the next step toward real-world, turbulent flow had yet to be completed – this was the plan for Phase II. Emphasis was placed on accuracy, efficiency and ease-of-use with the intention of being a credible component of a larger system. The capabilities after the Phase II included non-equilibrium and equilibrium chemistry for inviscid, laminar and turbulent flows on multi-zone, structured meshes.

During the two years of the project, the following technical objectives were accomplished:

1. Implemented sensitivity equivalents to the zero, one and two-equation turbulence modeling for real-world applications. These included algebraic (Baldwin-Lomax), one-equation (Spalart-Allmaras) and two-equation (Wilcox) models.

2. The sensitivity approach was developed for *wall-bounded*, turbulent shear flows to predict accurate profile and skin-friction sensitivities. Results using the three levels of turbulence modeling were compared to other existing methods: central-difference sensitivities and semi-empirical relationships.

3. Demonstrated accurate and efficient sensitivity computations for compressible, turbulent *free-shear* flows. We computed the sensitivity of a compressible shear layer to the convective Mach number defined as $M_c = (U_1 - U_2)/(a_1 + a_2)$. For a splitter-plate application, the sensitivity solutions successfully showed the trend of a diminishing growth rate in the shear-layer thickness as the convective Mach number increases.

4. Implemented implicit, zonal-boundary mapping to decrease multi-zone CPU. Most practical calculations about complex configurations require a multi-zone topology. This objective improved numerical performance for problems with zonal boundaries.

1.4 Relationship to Plus Up

The commercial product resulting from the Phase II effort was a robust multi-zone, *structured* sensitivity solver. However, the needs of the Air Force also include unstructured-grid applications particularly through the code, COBALT. The unstructured methodology has increased in popularity over the years with the need to study more and more complex geometries. To handle the present and future needs of the Air Force's design optimization requirements, this Plus-Up effort has focused on extending the sensitivity-equation method developed during a Phase II SBIR to unstructured meshes. The key features of the Plus-Up work include the abilities to

1. input grid and flow data from a COBALT solution,
2. efficiently solve the sensitivity equations for unstructured topologies,
3. compute sensitivity solutions on distributed-parallel architectures using the Message Passing Interface (MPI) and
4. use a graphical user interface to automate the solution process.

The development of this unstructured sensitivity capability provides the Air Force with a means to quickly compute flow sensitivities using COBALT as their base flow solver.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

Chapter 2

Project Objectives

The primary goal of this SBIR project has been to provide a robust, commercial sensitivity package compatible with any CFD flow solver – whether structured or unstructured. Phase I set the stage by focusing on accuracy and efficiency. High-order accurate sensitivity solutions were obtained more quickly and for multiple design variables during a single run. The Phase II focused on developing the software for real-world turbulent applications on structured grids. The software now has the capability to compute flow sensitivities for zero, one and two-equation turbulence models. The Plus-Up effort proposed an extension of the sensitivity capability to unstructured meshes on distributed architectures. A graphical user interface was proposed to further reduce user workload.

To achieve these goals, the following technical objectives were identified:

Objective 1: *Solve the sensitivity equations on unstructured meshes.*

We will develop a computational tool for computing flow and geometric sensitivities consistent with the discretization and physical modeling in Cobalt. Issues to address include: flow-sensitivity solution handshake, grid/solution I/O, SEM boundary conditions, flux and source-term evaluation, time integration and output.

Objective 2: *Incorporate MPI for distributed parallel capability.*

AeroSoft's unstructured flow solver, $\mathcal{GUST}$, utilizes MPI to perform distributed parallel simulations. The advantages of the message passing model are platform portability, an ability to expand beyond the number of processors available on any single computer, and easier debugging than shared parallel since each process has access to its own local memory. Using MPI, we will develop distributed parallel capability in the unstructured version of $\mathcal{SENSE}$.

Objective 3: *Provide a graphical user interface.*

Graphical User Interfaces (GUIs) are an important part of the current generation of commercial CFD grid generators and flow solvers. We will incorporate a GUI that controls calls to the flow and sensitivity solvers. This objective allows the sensitivity solver to remain a standalone product for steady calculations or partner with any flow solver for unsteady sensitivities.

Objective 4: *Provide final report, binary software and complete documentation.*

A final report will be furnished containing the project objectives, work performed, results obtained, and estimates of technical feasibility.

The binary software will be provided with free upgrades and bug-fixes for two years following the closing date. A comprehensive manual will be provided with the software as well.

Chapter 3

Work Performed

3.1 General Sensitivity Formulation

The idea of a flow sensitivity is that of a linear approximation – a partial derivative of the flow with respect to a parameter of interest. Here we use the symbol η to denote a generic parameter. To emphasize that the flow solution depends on position, time and the parameter, we write

$$\mathbf{Q}(x, y, z, t;\ \eta).$$

The sensitivity we seek is then formally given by

$$\mathbf{S}(x, y, z, t;\ \eta) \equiv \frac{\partial \mathbf{Q}}{\partial \eta}.$$

Our objective is to derive the linear sensitivity equation for $\mathbf{S}(\cdot;\eta)$. We do this formally, by differentiating the partial differential equations corresponding to each turbulence model.

Three-dimensional flow of a viscous fluid is governed by a system of non-linear, hyperbolic partial differential equations which can be written in integral form as

$$\frac{\partial}{\partial t}\iiint \mathbf{Q}\, dV + \oint_A \left(\mathbf{F}(\mathbf{Q}) \cdot \hat{\mathbf{n}}\right) dA = \oint_A \left(\mathbf{F}_v(\mathbf{Q}) \cdot \hat{\mathbf{n}}\right) dA + \iiint \mathbf{W}\, dV, \tag{3.1}$$

subject to boundary conditions on the surfaces surrounding a defined control volume. The conservative, mean-flow variables are $\mathbf{Q} = \mathbf{Q}\,(x, y, z, t) = [\rho, \rho u, \rho v, \rho w, \rho e_0]^T$ and represent the mass, momentum and total energy per unit volume of the fluid. Additional transport equations are needed for one and two-equation models where the conservative variables depend upon the model ($\rho\tilde{\nu}$ for Spalart-Allmaras, ρK and $\rho\omega$ for Wilcox). The surface integrals represent the inviscid and viscous fluxes ($\mathbf{F}$ and $\mathbf{F}_v$); the source term represents production and dissipation of the conserved turbulence variables. This integral formulation is a fundamental starting place for the finite-volume discretization [1, 4] utilized in the $\mathcal{SENSE}$ software.

As presented in this report, the sensitivity equations are derived by first writing Eqn. (3.1) as a partial differential equation in Cartesian coordinates. We formally differentiate this PDE with respect to the design variable (η), and then interchange the order of differentiation between η and the temporal/spatial dimensions. The resulting PDE is then recast as an integral conservation equation for finite-volume applications. The sensitivity equations are

$$\frac{\partial}{\partial t}\iiint \mathbf{S}\,dV + \oint_A \left(\mathbf{F}'(\mathbf{Q},\mathbf{S})\cdot\hat{\mathbf{n}}\right)\,dA = \oint_A \left(\mathbf{F}'_v(\mathbf{Q},\mathbf{S})\cdot\hat{\mathbf{n}}\right)\,dA + \iiint \mathbf{W}'(\mathbf{Q},\mathbf{S})\,dV \qquad (3.2)$$

where $(\cdot)'$ represents differentiation with respect to η. An important advantage of the sensitivity-equation approach is that the governing sensitivity equations (*i.e.,* Eqn. (3.2)) are linear in the sensitivity, $\mathbf{S}$, via chain-rule differentiation. The flow solution $(\mathbf{Q}(\cdot,\eta))$, which satisfies Eqn. (3.1), enters through spatially varying coefficients. In practice, the numerical solution to this linear problem is obtained with relatively little computational cost.

3.2 Development of an Unstructured Sensitivity Solver

The Plus-Up effort has focused on developing an unstructured sensitivity-equation solver which is compatible with Cobalt. Cobalt has been developed by the Air Force under the CFD Computational Technology Area (CTA) of the High Performance Computing (HPC) Common High-Performance-Computing Software Support Initiative (CHSSI). Cobalt is an unstructured-grid, finite-volume code that solves the compressible Euler/Navier-Stokes equations using Godunov's method assuming an ideal-gas law. Arbitrary cell types are permitted which include triangles, quadrilaterals, tetrahedra, prisms, pyramids, and hexahedra. Emphasis is placed upon generality, robustness, accuracy and ease-of-use so that Cobalt can be a building block for interdisciplinary CFD efforts.

To develop the sensitivity software, we begin with the programming framework set forth in AeroSoft's unstructured flow solver, $\mathcal{GUST}$. Since flow and sensitivity solvers both solve integral governing equations, the computational steps required of our unstructured sensitivity solver are very similar to those of a flow solver. That is, much of the coding effort spent developing $\mathcal{GUST}$ can be utilized in the unstructured sensitivity solver. This includes grid and solution I/O, time-integration schemes, non-dimensionalization, metrics, message passing, partitioning and data structures among others. Cobalt grids are read into the software through provided grid formats. The solution is using a compatible Fieldview format.

Even though some of the code was taken from the $\mathcal{GUST}$ flow solver, a significant portion of the solver is specific to the sensitivity equations. Major parts of the sensitivity development included: boundary conditions for both flow and sensitivity variables, flux sensitivities and Jacobians, routines for MPI, and a post-processing utility to compute variables which depend on the sensitivities. Additionally, a central-difference utility was created which generates central-difference sensitivities for comparison to the SEM results.

3.3 Inviscid Sensitivities

A simple diamond airfoil with wedge angle of 8.531° is used to verify the inviscid-sensitivity software. The diamond airfoil penetrates a Mach 2.0 free stream with standard air properties at sea-level. For this test case, we examine the sensitivities to the free-stream x-component of velocity, U_∞. For the central-difference solution, the free-stream velocity was perturbed 10% above and below the baseline velocity. Figure 3.1 shows the density sensitivity computed by the central-difference method and the sensitivity solver. Figure 3.2 shows the pressure sensitivity. Both sensitivities have been scaled to the same range. The sensitivity solver

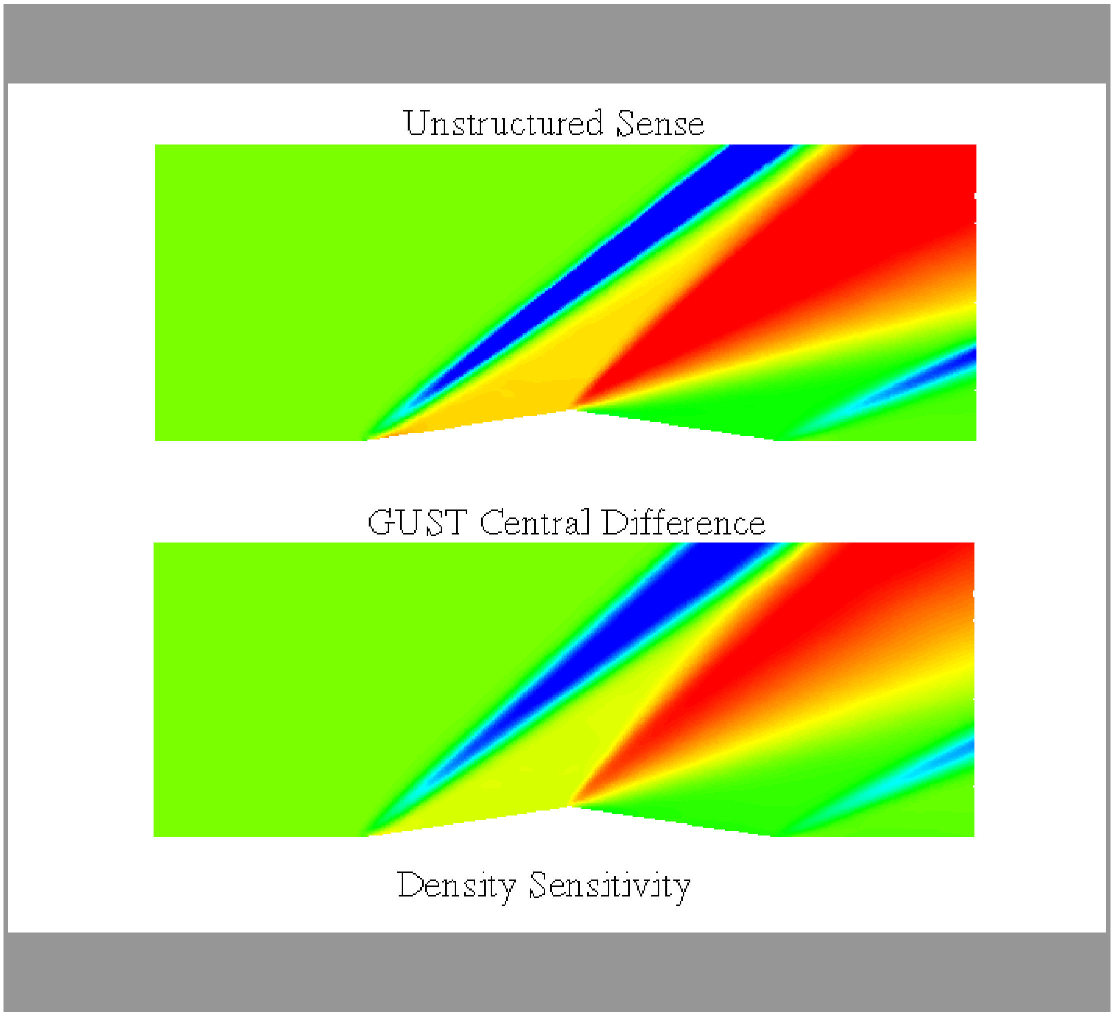


Figure 3.1: Sensitivity of density to free-stream velocity. Comparison between SEM and CD.

yields slightly larger maximums and slightly smaller minimums than the central-difference method. The pressure sensitivity on the lower boundary is given in Figure 3.3. Both the SEM and central difference agree well with shock-expansion theory. This simple case serves as one validation case for an inviscid, sensitivity problem.

3.3.1 Viscous Terms

In an unstructured solver, at least two methods exist for computing viscous gradients. One is based on a linear gradient formulated by Barth [2]. The other is based on a linear K-Exact reconstruction and the derivatives of the reconstruction [3]. The Jacobian of the gradients of the sensitivity variables is significantly more difficult for Barth gradients than for the K-Exact gradients. However, in an unstructured flow solver, the Barth gradients are more robust than the K-Exact method.

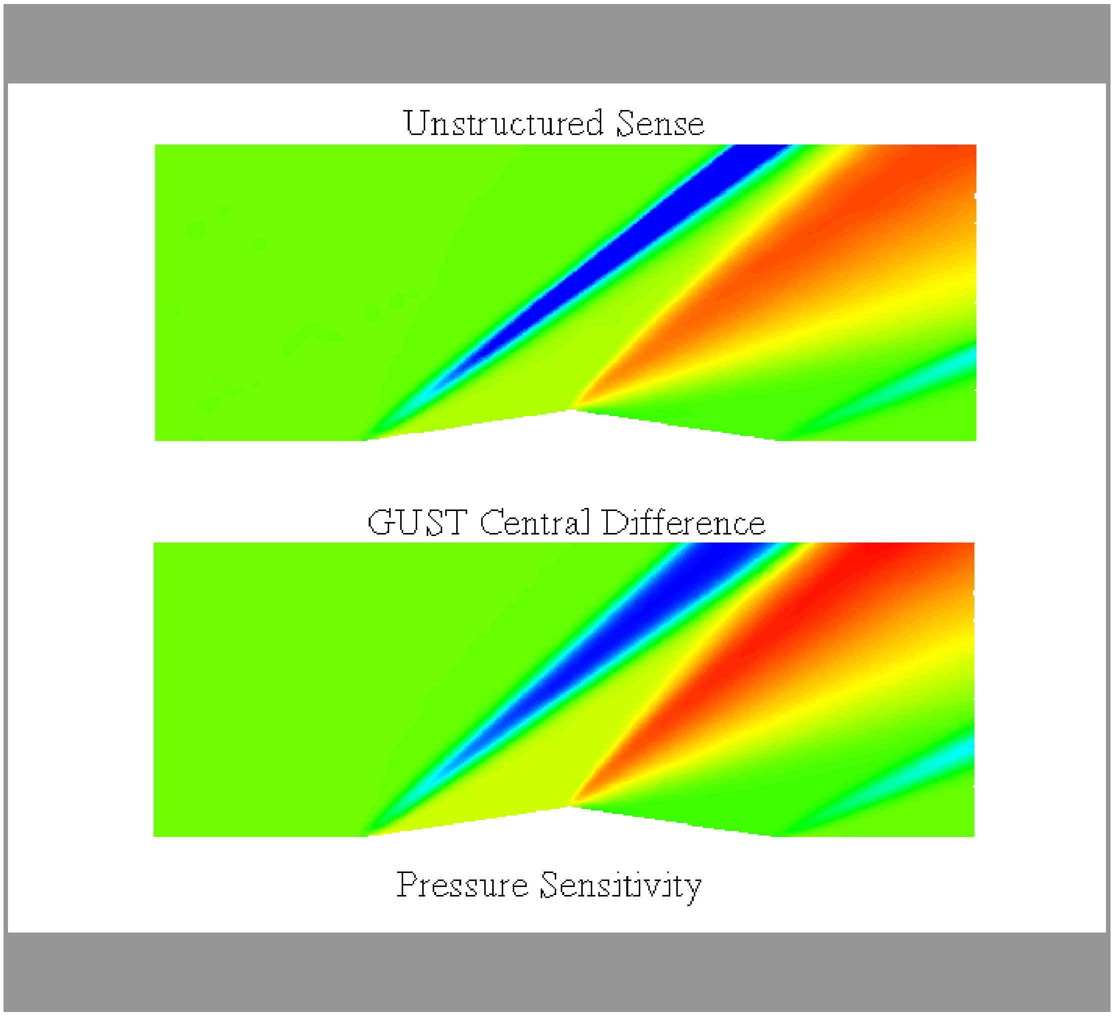


Figure 3.2: Sensitivity of pressure to free-stream velocity. Comparison between SEM and CD.

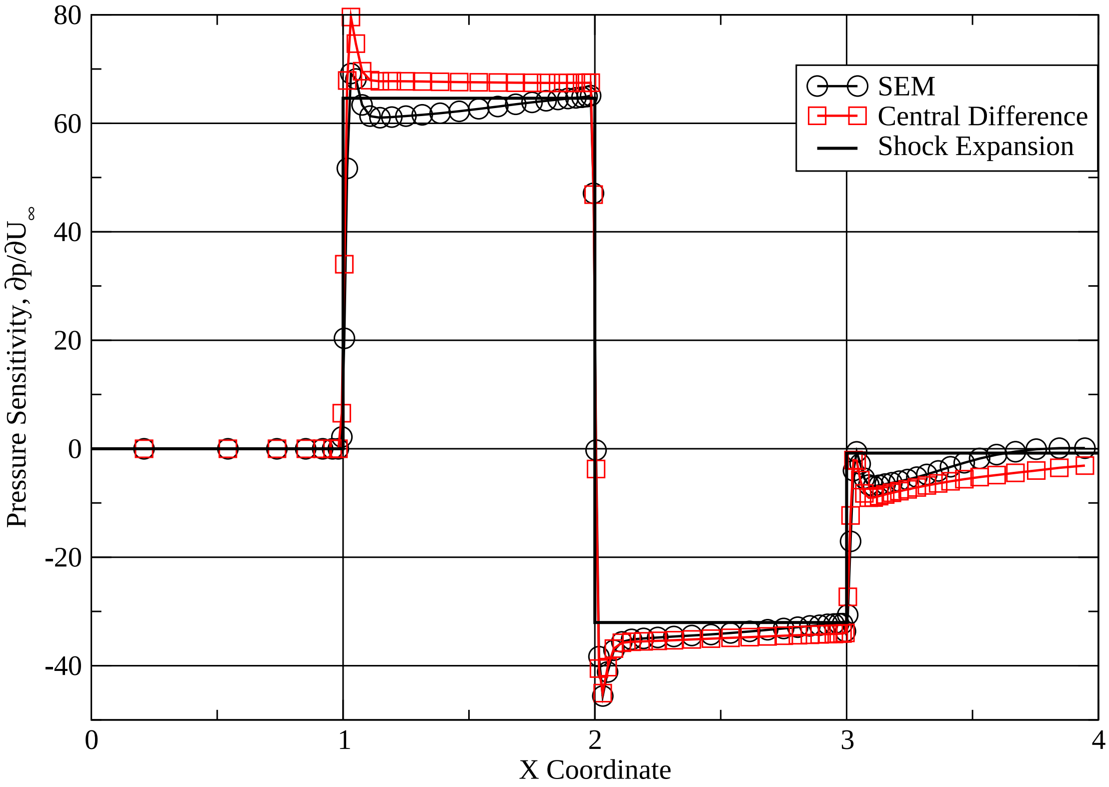


Figure 3.3: Pressure sensitivity along the diamond airfoil compared to shock-expansion theory.

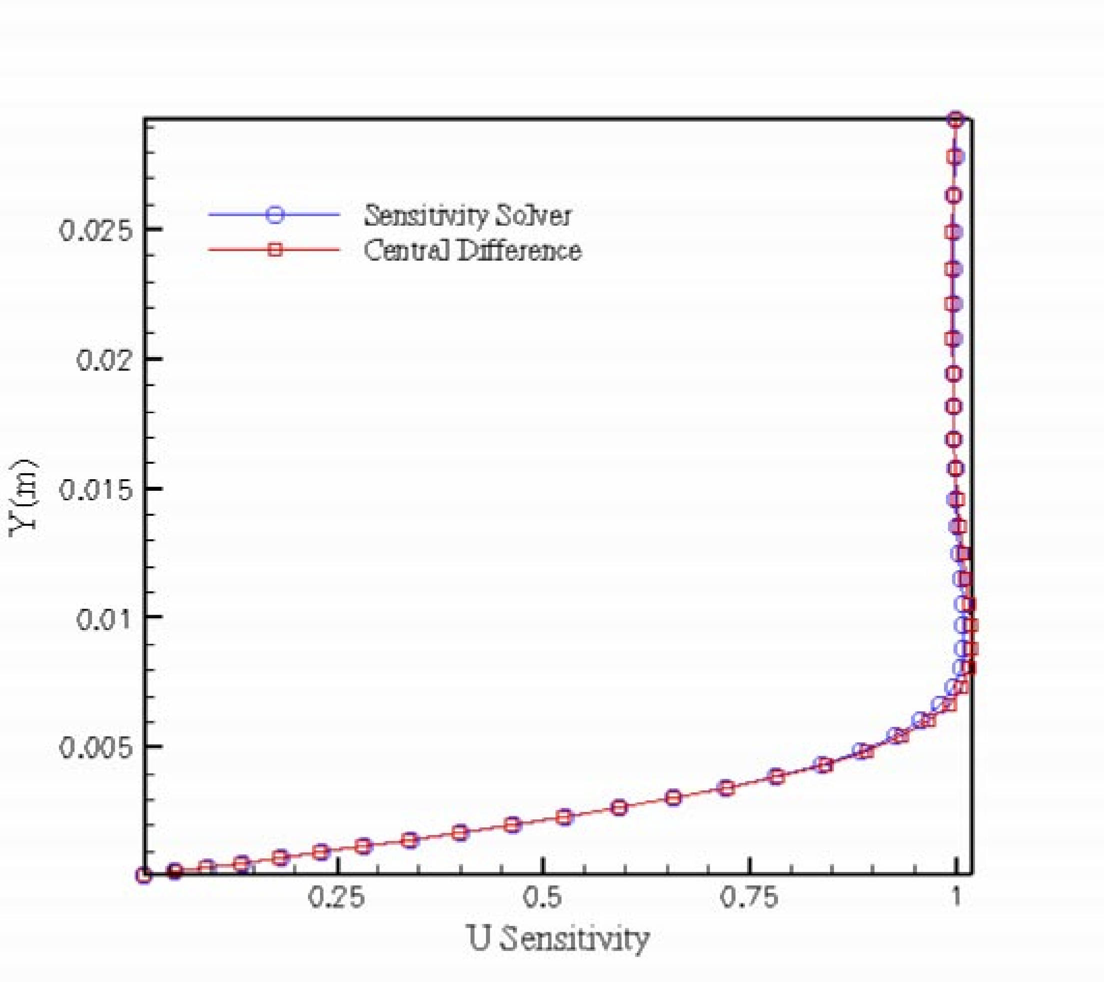


Figure 3.4: Sensitivity of the laminar, flat-plate velocity profile to the edge velocity as compared to a central difference.

Flat-Plate Results

Two problems were run to test the viscous formulation. The first simulates the laminar boundary layer formed by a flat plate. The plate measures $L = 1\ m$ in length in a $M_\infty = 0.3$ flow field. The Reynolds number for this problem is $Re_L = 200,000$. The results at the end of the plane are provided in Figure 3.4. In this figure, the sensitivity of the x-component of the velocity is shown versus the normal distance from the plate. The sensitivity solver agrees well with the central-difference method. Both gradient methods were run and both provide good agreement. Shown in this figure is the K-Exact gradient method.

The $K-\omega$ turbulence model developed during the Phase II was implemented and tested on the flat plate as well. Figure 3.5 shows the sensitivity of the law of the wall for a Reynolds number of $Re_L = 2 \times 10^6$. The results in the figure imply that the sensitivity solver produces good agreement compared to theoretical inner and log-layer results.

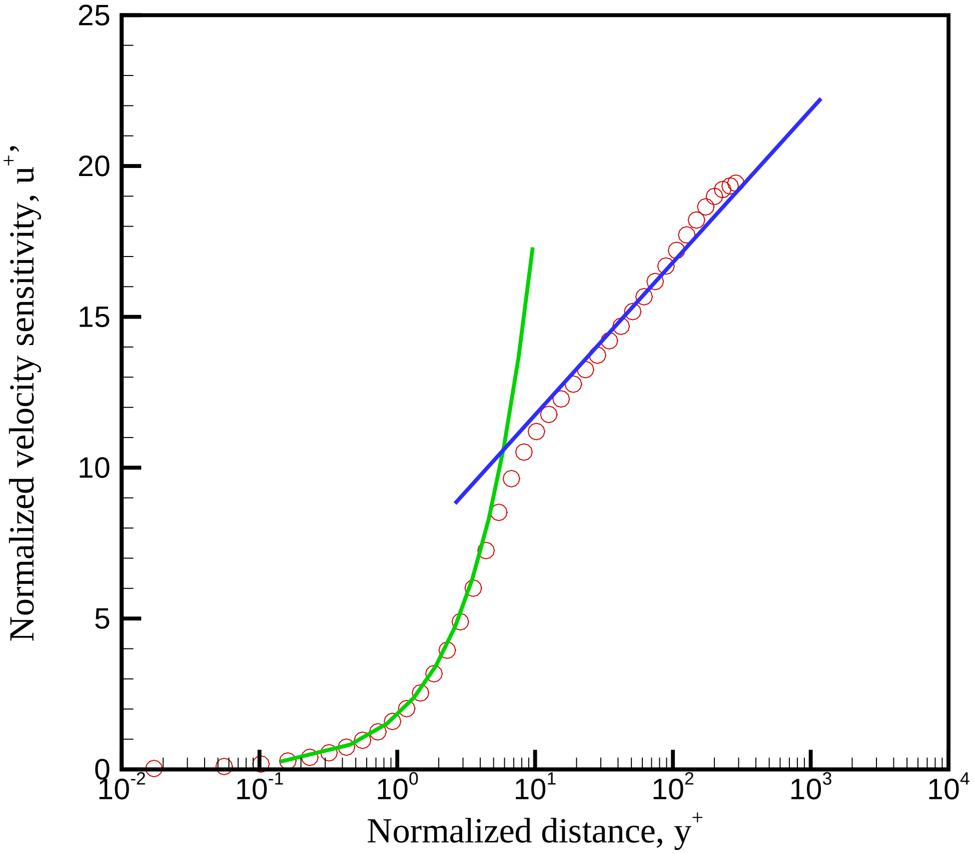


Figure 3.5: Two-equation sensitivity of the law of the wall.

Analytic Forebody

The second viscous test case is an analytic forebody. The generic forebody has a surface geometry which is known analytically and experimental data exists for several Mach numbers and angles of attack. The conditions for this investigation are $M_\infty = 1.7$ and $\alpha = 0°$. The design variable is the free-stream velocity, U_∞. The mesh contains 154,408 cells and the case was run on 4 Athlon XP2200's in 8 minutes 8 seconds (wall clock). This represents 4.4% of the flow CPU time. Pressure and velocity sensitivity results are provided in Figure 3.6 and Figure 3.7. The figure on the left is the result from the sensitivity solver. The figure on the right is the result from the central-difference method. The pressure sensitivity ranges from $p'/p_\infty \in (-38.6, +37.6)$. Again, the sensitivity solver and central-difference agree well. Figure 3.7 shows the u-velocity sensitivity.

3.4 Implementation of Distributed Parallel via MPI

The unstructured sensitivity solver utilizes MPI for performing distributed parallel simulations. In the message passing model, each process has only local memory and communicates with the other processes by sending and receiving messages. The defining feature of the message-passing model is that data-transfer from the local memory of one machine to the local memory of another machine requires operations to be performed by both the sending machine and the receiving machine. The advantages of the message passing model are portability to different platforms, ability to expand beyond the number of processors on any single computer, and easier debugging since each process has access to its own local memory.

By their nature, message passing algorithms do not globally share data. This means that one processor does not have direct access to information on another processor. The unstructured sensitivity solver implements a method of breaking the domain into smaller subsets, called partitions. The method is similar to breaking a computational domain into zones. All of the data structures are local to a given partition, a requirement of message-passing algorithms. A partition boundary exists between adjacent partitions similar to a zonal boundary. All data transfers between partitions occur at the partition boundary.

The parallel algorithm implemented in the unstructured code has proven to be very successful. As a benchmark, timings for the flow around a business-class jet are obtained. The mesh shown in Figure 3.8 contains 361,996 cells and the free-stream conditions for the case are $M_\infty = 0.7$ and $\alpha \approx 10°$. On a single processor R10000 SGI, the CPU time per iteration is 48.01 seconds. For 16 processors the time per iteration is 3.01 seconds, almost a perfect 16 times speed up. As the machine limits are approached, the efficiency does decrease. At the machine limit of 64 processors, the code requires 1.03 seconds per iteration – an approximately 47 times speed up. The memory requirements are slightly higher than a single processor job. However, the memory is distributed across the different machines. A single processor requires approximately 920 megabytes of memory for the jet. A two processor simulation requires 465 megabytes per processor (roughly half).

3.5 Developing a Graphical User Interface

Several layers of software exist in any graphical user interface. At the instruction level sits the C programming language which was first implemented by Dennis Ritchie in the UNIX

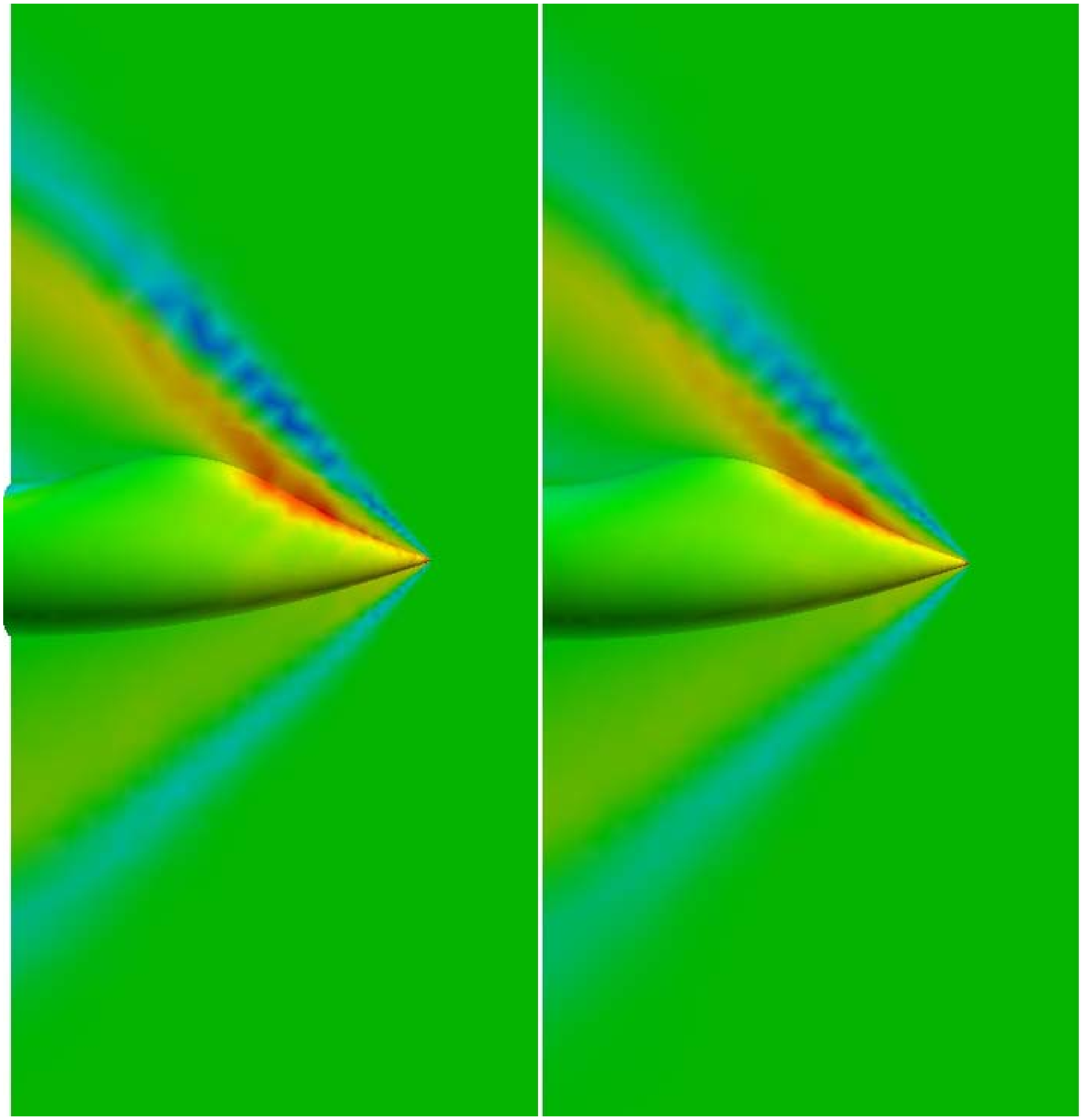


Figure 3.6: Sensitivity of the forebody pressure to the free-stream velocity (left) as compared to a central difference (right).

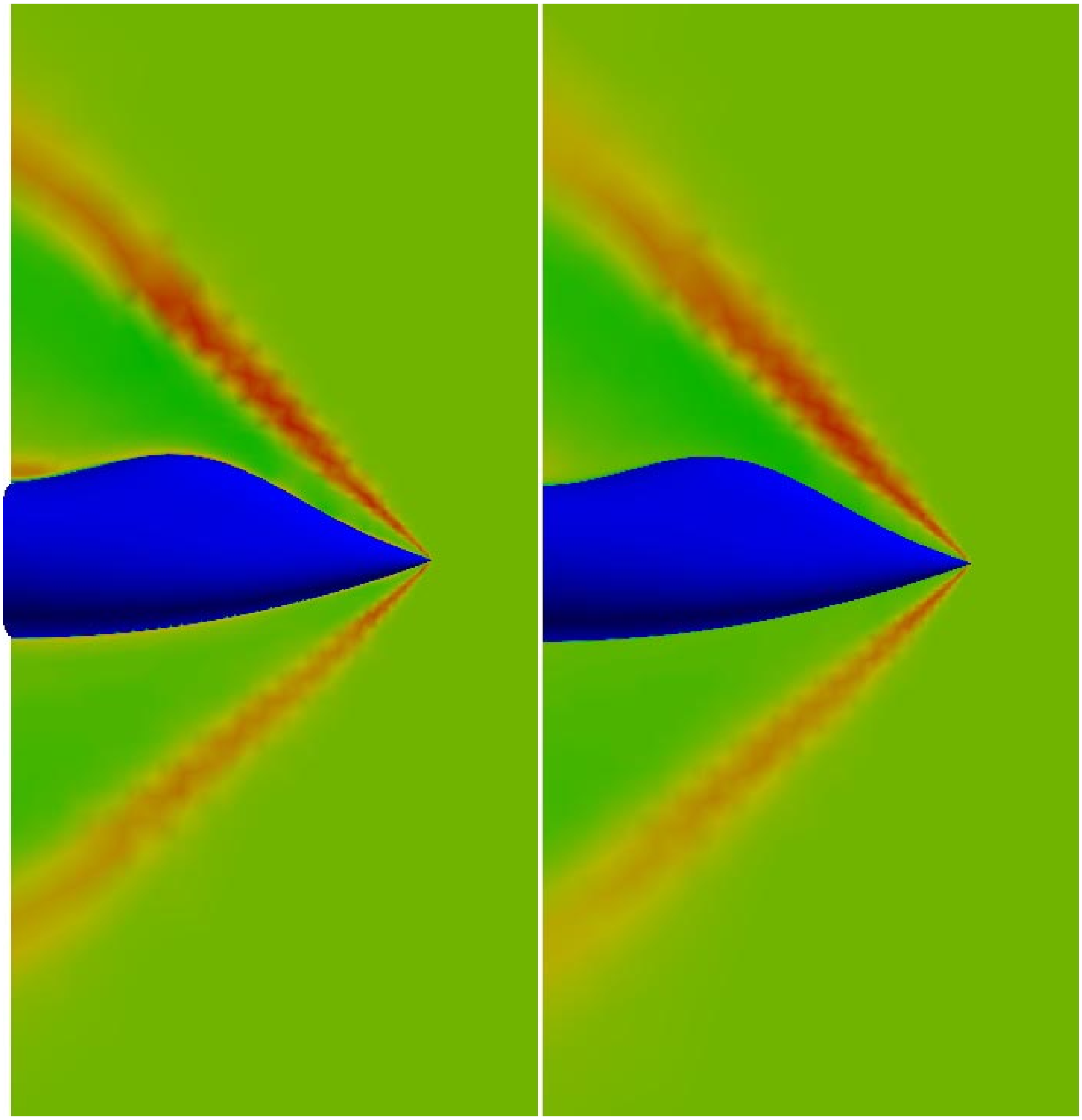


Figure 3.7: Sensitivity of the forebody x-component of velocity to the free-stream velocity (left) as compared to a central difference (right).

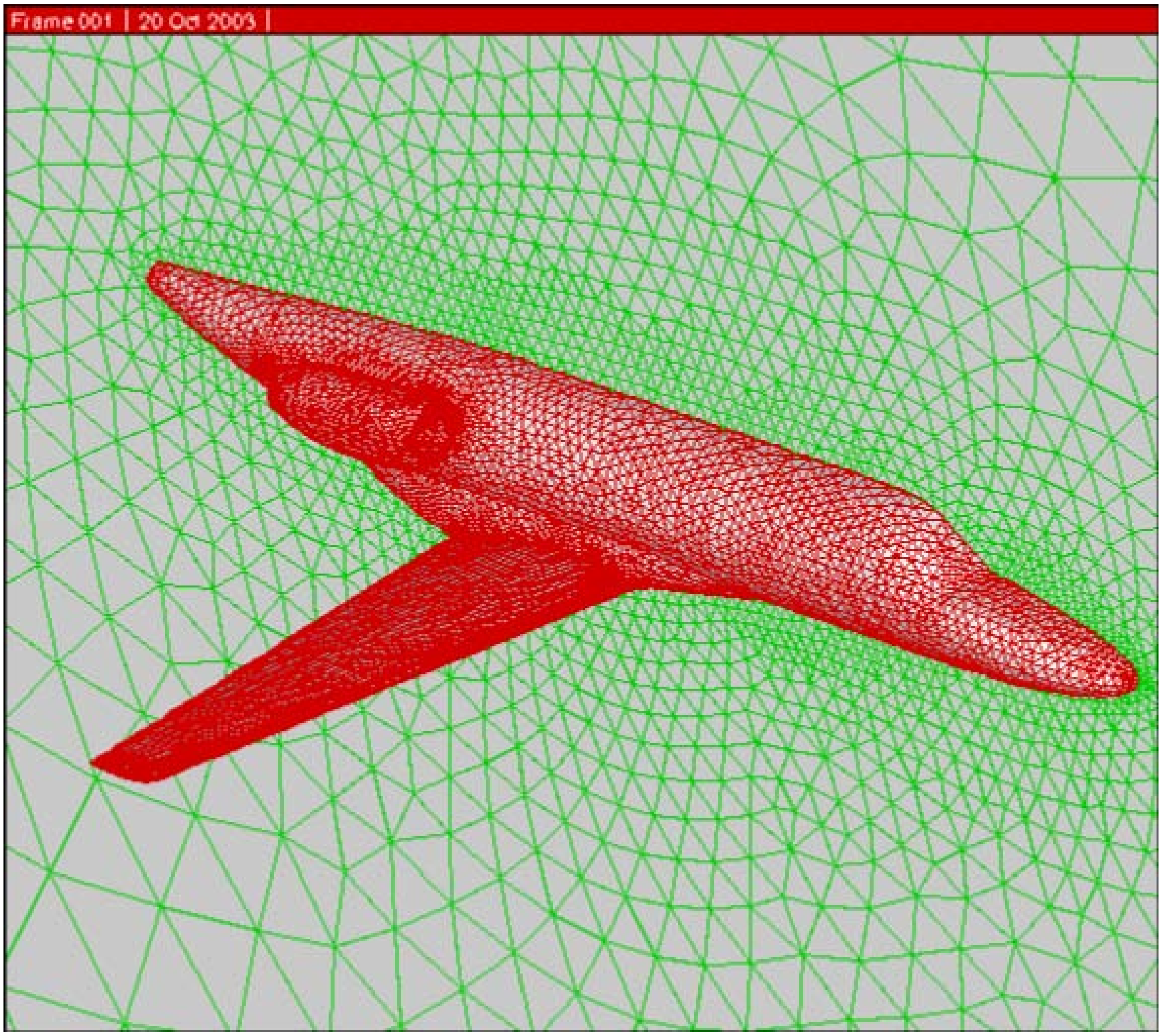


Figure 3.8: Mesh around a business jet.

operating system. The ANSI C standardization in 1989 allowed for C applications to be portable across different platforms without encountering compiler discrepancies.

A programmer in C can call both library functions and system functions. Several important libraries for the interface designer include: `glib`, X, GDK, GTK+, and GNOME. The `glib` library of functions contains many of the most important functions for a Linux application. In particular, the main-loop function allows handling multiple resources while executing functions in the application. The X graphics library contains low-level functions which control the display. These include window display, responding to mouse movements and clicks, window movement and re-sizing. GDK (GIMP Drawing Kit) simplifies the programmer's access to the simple tasks contained in X. GDK organizes the small, individualized tasks of the X library into one of two function calls. The GIMP Toolkit (GTK+) organizes the GDK functions into objects such as buttons, labels, text windows. These objects are called widgets. The GNOME library is a specialization of the GTK+ library. With a single GNOME call, a complete pop-up dialog box can be displayed which may contain many GTK+ widgets.

To develop the user interface, we use the GNOME/GTK+ toolkit, a standard toolkit for creating window applications in the Linux operating system. However, these applications are portable to most UNIX systems including Cray, SGI, HP, and SUN.

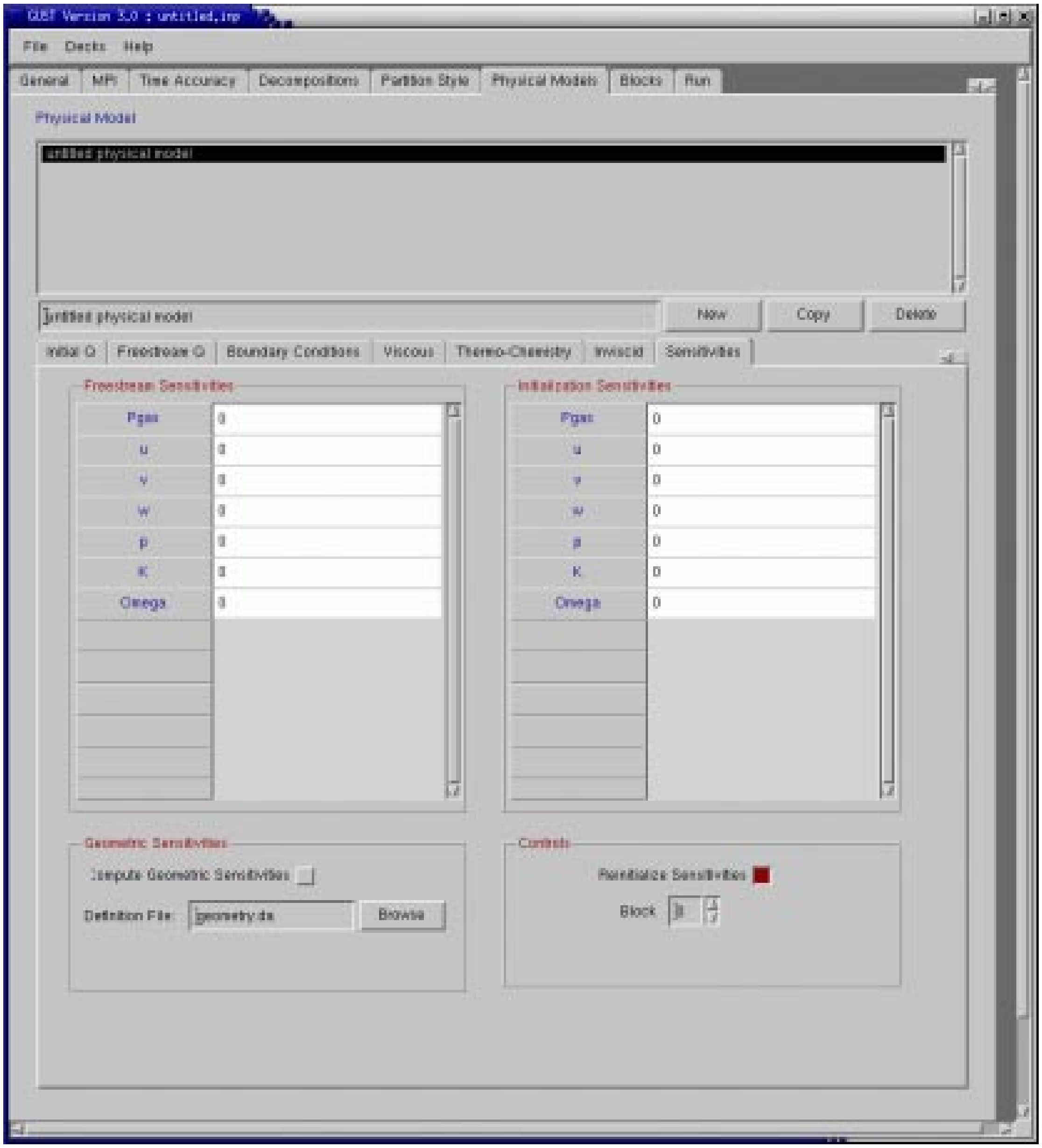
Several aspects of the interface are specific to the sensitivity problem. For example, free-stream specification of turbulence-model sensitivity variables is added and shown in Figure 3.9. Since not all turbulence models are supported, error handling is added which warns the user if an unsupported model is used for sensitivity calculations. For example, Spalart-Allmaras model is implemented for the flow equations but not for the sensitivities. Similar warnings are handled for the certain transport and thermodynamic models which are not supported at this time (*e.g.,* non-equilibrium vibration, Gupta transport). In addition, support for reinitializing the sensitivities is now available to the user in the front end. The sensitivity solver must also work without the user implementing the interface, so redundant error-handling support is added to ensure that the physical models selected are available to the solver.

A "Run" tab is added to the user interface to allow the user to run the flow solver, sensitivity solver, decompose the grid, launch the post processing GUI, and post process the results. In addition, residual plotting and cluster information are provided. Figure 3.10 shows this "Run" tab. The top table shows the MPI booted nodes, along with the 1 minute, 5 minute, and 15 minute load averages as well as the total and free memory on each node. The number of required processes is shown under the table. The user can select which nodes to run by selecting the "RunOn" button. The buttons at the bottom allow the user to launch the flow solver or sensitivity solver from within the front-end.

The graph below the node table shows a plot of the residual for either the flow solver or the sensitivity solver. You can manually refresh or have the interface automatically refresh the graph every three seconds.

3.6 Conclusions and Future Work

As a result of this SBIR effort, a concept for computing sensitivities has grown into two commercial products – one for multi-zone, structured grids and another for multi-partitioned,



untitled physical model

Freestream Sensitivities

Pgas	0
u	0
v	0
w	0
p	0
K	0
Omega	0

Initialization Sensitivities

Pgas	0
u	0
v	0
w	0
p	0
K	0
Omega	0

Geometric Sensitivities

Compute Geometric Sensitivities □

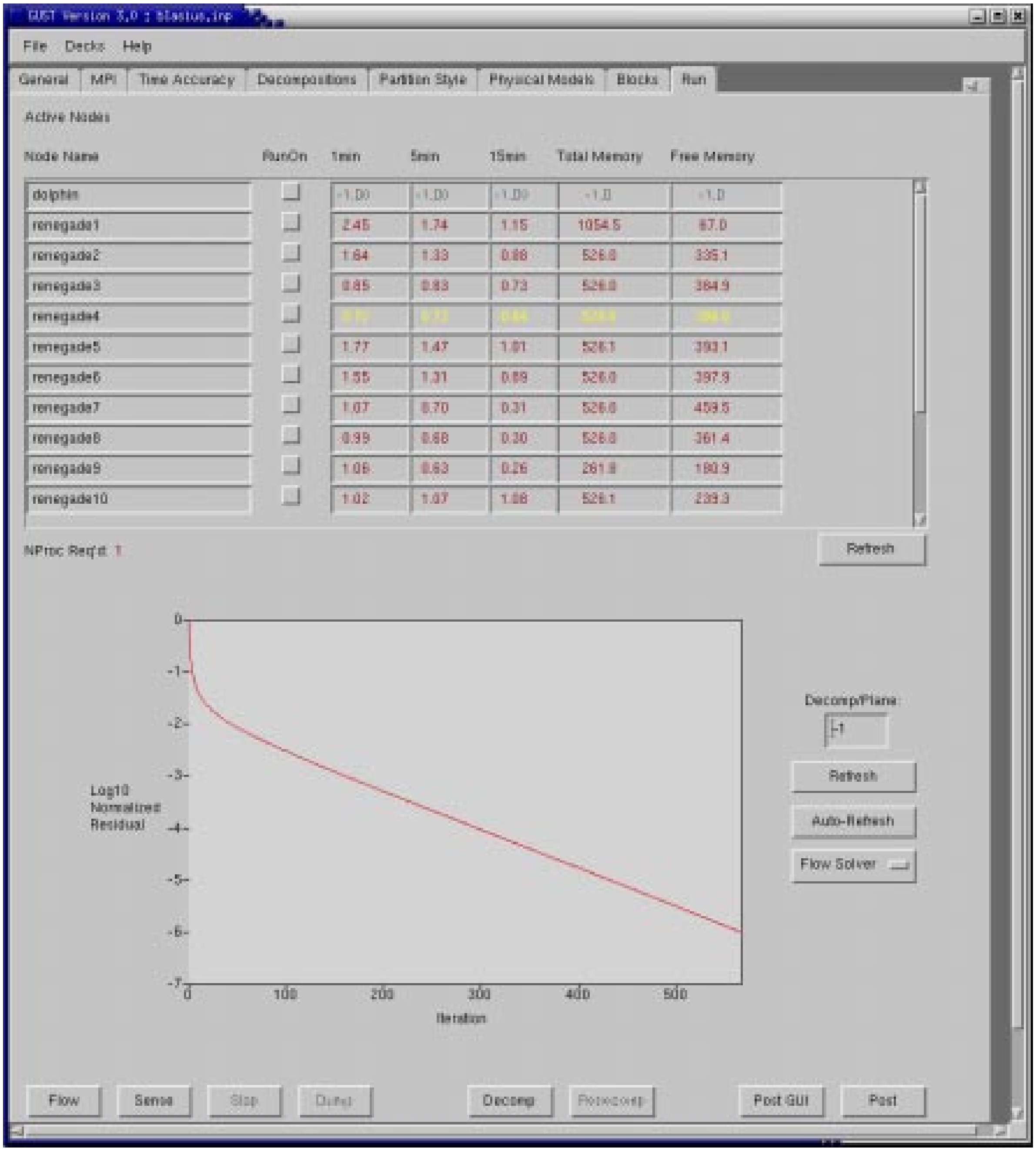
Definition File: geometry.da

Controls

Reinitialize Sensitivities ■

Block 0

Figure 3.9: The user interface section relevant to turbulence modeling.



Node Name	RunOn	1min	5min	15min	Total Memory	Free Memory
dolphin	□	-1.00	-1.00	-1.00	-1.0	-1.0
renegade1	□	2.45	1.74	1.15	1054.5	67.0
renegade2	□	1.64	1.33	0.88	526.0	335.1
renegade3	□	0.85	0.83	0.73	526.0	384.9
renegade4	□	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
renegade5	□	1.77	1.47	1.01	526.1	393.1
renegade6	□	1.55	1.31	0.89	526.0	397.9
renegade7	□	1.07	0.70	0.31	526.0	459.5
renegade8	□	0.99	0.68	0.30	526.0	361.4
renegade9	□	1.06	0.63	0.26	261.9	180.9
renegade10	□	1.02	1.07	1.08	526.1	239.3

NProc Req'd: 1

Decomp/Plane: -1

Figure 3.10: Portion of the graphical user interface showing the MPI nodes and residual history.

unstructured grids. Both sensitivity solvers compute sensitivities to design variables in less than 10% of the time required for computing the baseline flow solution. The sensitivity equations have been extended from simple inviscid flows to turbulent, chemically reacting flow applications. Design variables can be either flow-related or geometry-related. However, problems with geometric design variables in turbulent flows still persist and work continues to resolve those issues. The primary problem is that of computing an accurate flow gradient on a highly stretched mesh. Because of the tight spacing near solid boundaries, any small perturbations in the flow variables are amplified in the gradient. Erroneous flow gradients lead to poor sensitivity solutions when the design variable is related to the geometric shape. A least squares fit of the near-wall data has not led to more accurate results. At the current time, we are investigating a non-linear, heat-equation solution with additional high-frequency noise. Resolving this difficulty is the current focus of the sensitivity work.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

Bibliography

[1] Inc. AeroSoft. **GASP Version 3 User's Manual**. AeroSoft, Inc., 1996.

[2] T. J. Barth. "Numerical Aspects of Computing Viscous High Reynolds Number Flows on Unstructured Meshes". AIAA Paper 91-0721, 29^{th} *AIAA Aerospace Sciences Meeting*, January 7-10, 1991.

[3] T. J. Barth. "Recent Developments in High Order K-Exact Reconstruction on Unstructured Meshes". AIAA Paper 93-0668, 31^{st} *AIAA Aerospace Sciences Meeting and Exhibit*, January 11-14, 1993.

[4] C. Hirsch. **Numerical Computation of Internal and External Flows, Volume 1**. John Wiley and Sons, New York, 1988.

[5] AFRL/VAAC Web Site. `http://www.va.afrl.af.mil/COE/CompS/comps_index.html`. Mission Statement.